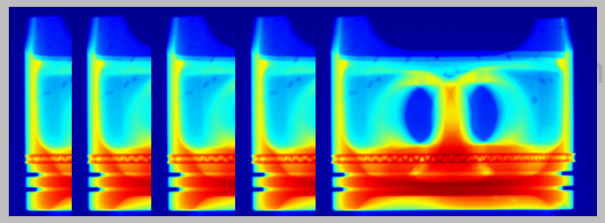
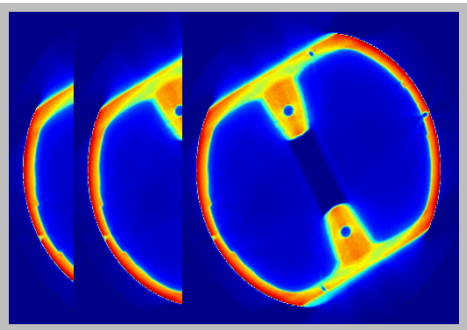
SAND2014-20042PE

Exceptional service in the national interest

Sandia National Laboratories





Irregular Big Data Computed Tomography on Multiple Graphics Processors Improves Energy-Efficiency Metrics for Industrial Applications

Dr. Edward S. Jimenez

Team: Dr. Edward S. Jimenez, Laurel J. Orr, Kyle R. Thompson, Dr. Ryeojin Park, Eric L. Goodman

Hispanic Engineers National Achievement Awards Conference 2014

October 2014

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

Computed Tomography

- Computed Tomography (CT) is an indirect 3D imaging technique.
- Input: Set of X-ray images acquired about a center of rotation.
- Output: Three-dimensional approximation of internal and external structure
- Reconstruction: Convolution- Backprojection Algorithm (Feldkamp-Davis-Kress)
- Geometry and Configuration of CT System determines magnification
- Reconstruction algorithm is $O(n^4)$

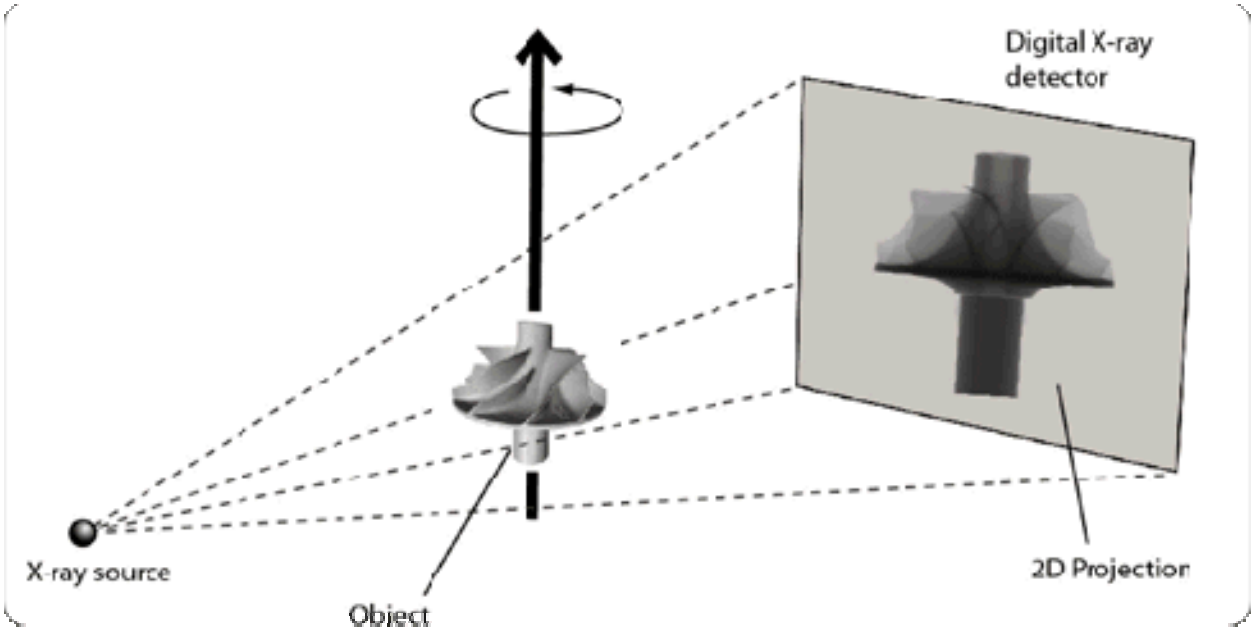


Image Source: http://www.xviewct.com/assets/images/how-ct-works.gif

GPU

Sandia National Laboratories

- Graphics Processing Units are coprocessors that handle image manipulation and now are being used for general purpose computing.
- Capable of Teraflops!
- This massive computational capability of GPUs can be harnessed for many applications.
  - Parallel computing environment
  - Fast dedicated memory
  - Fast Cache
- CT Reconstruction from projection images requires many arithmetic and trigonometric operations for every volumetric pixel (voxel).

CT on GPUs

Sandia National Laboratories

- "Porting" CT reconstruction on GPUs has shown major bottlenecks.
  - Usually not an issue with medical datasets.
  - Memory uploads/downloads to device (GPU).
  - What ratio of x-ray data to volume should be allocated?
- Traditional CPU-based code reconstructed one slice at a time
  - Predicable memory access even when multi-threaded.
- GPU-based reconstruction
  - Massively multithreaded environment creates scattered memory reads if large x-ray data is utilized.
  - Scattered Memory reads severely hinders performance!
  - Suddenly reconstruction becomes an Irregular Problem!

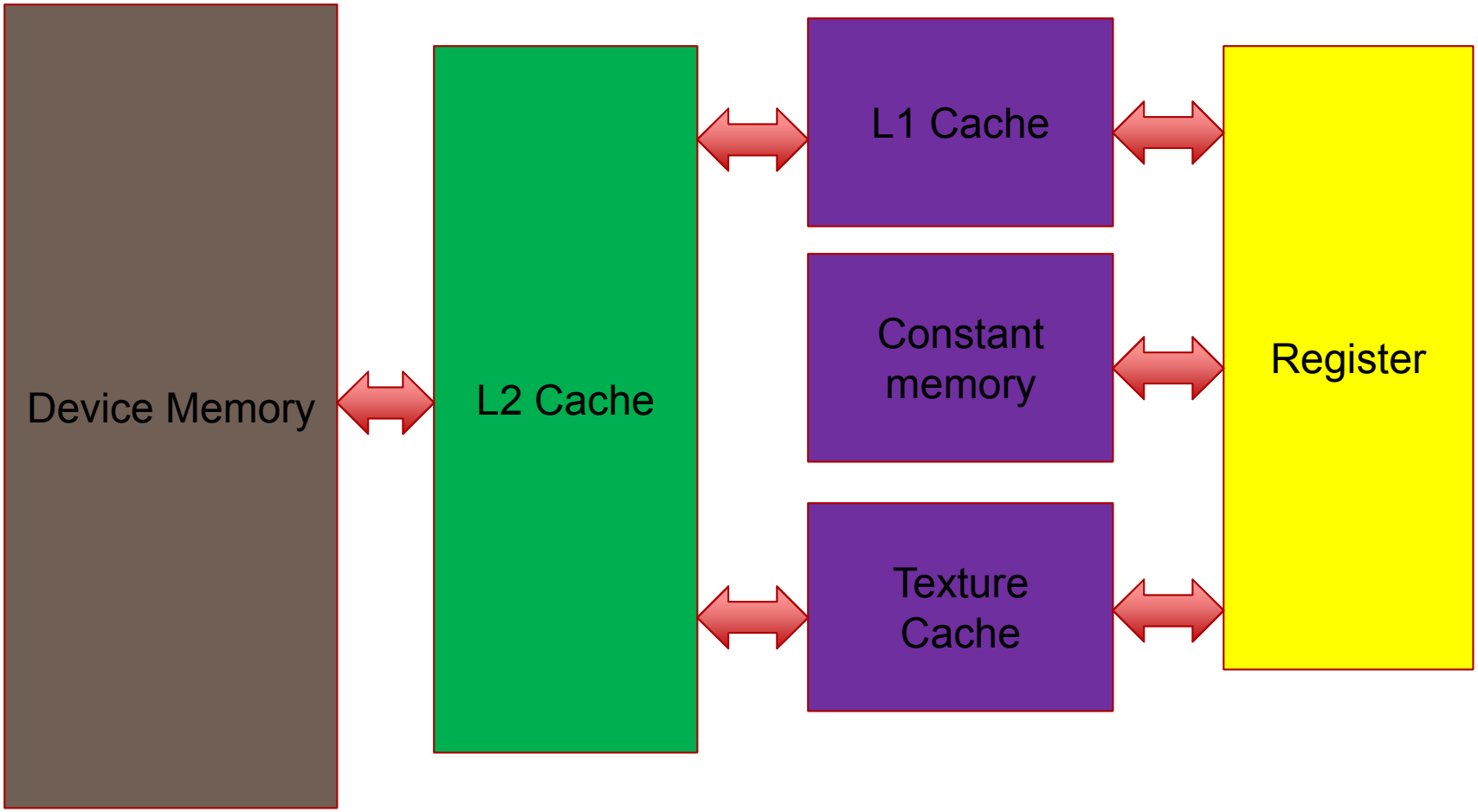
Approach

Sandia National Laboratories

- Maximize resources by blocking x-ray data and sub-volumes.
  - Not necessarily a new idea…
- Counter Intuitive: *Maximize* x-ray data uploads to device!
  - Partition x-ray images and batch small x-ray image subsets
- Volume: Use most GPU memory for direct volume storage.
- Utilize GPU-Specific Hardware/Features
  - Massive parallelism
  - Texture memory/Texture Cache
  - Constant Memory
  - Data prefetch to pinned memory for fast upload
- Dynamic Task Partitioning
  - Ensures only relevant data is fetched and uploaded to the device

GPU Cache Hierarchy

Sandia National Laboratories



Implementation

Sandia National Laboratories

- CUDA Programming environment and C++
- Minimum requirements
  - Fermi-based architecture or newer
  - 1 GB device memory
  - At least one x-ray sub image and one image plane must fit together on device
- Allows for 1 – 8 GPUs per node
- Dynamic Partitioning determined by *slice-to-texture ratio* (*STR*)
- STR may not always be satisfied:
  - Resource maximization vs. Awkward task size
  - Reconstruction size – Too large or small?
  - Tail-end reconstruction

Dynamic GPU Tasking

Sandia National Laboratories

- For a given subvolume the amount of x-ray data necessary varies
  - Due to the geometry of the system.
  - Taken into account with STR to determine memory data allocation on device.
  - Typically, reconstruction along center slices require less data.

- Using OpenMP 2.0, a CPU thread controls one GPU in the system
  - Each GPU will usually be reconstructing sub-volumes of varying size
  - Load balancing difficult if subvolume is fixed for all GPUs

- No synchronization necessary for CPU threads while algorithm is executing.

- No synchronization necessary between GPU threads either.

- One atomic operation to update reconstruction progress and determine next subvolume to reconstruct.

FDK Kernel Layout

Sandia National Laboratories

- **Input:** X-ray data, index, and size, subvolume data, index, and size, system geometry
- Get thread *ID* and voxel positions $p_1, \ldots, p_s$ based on *ID*
  - **if** Thread *ID* position within ROI **then**
    - **for** Every slice *j* in slice block **do**
      - Set register value to zero
      - **for** Every image *i* in image subset **do**
        - » Determine texture interpolation coordinate in image *i*
        - » Update register value with texture fetch and scaling
      - **end for**
      - Update voxel p_j in global memory with register value
    - **End for**
  - **End if**

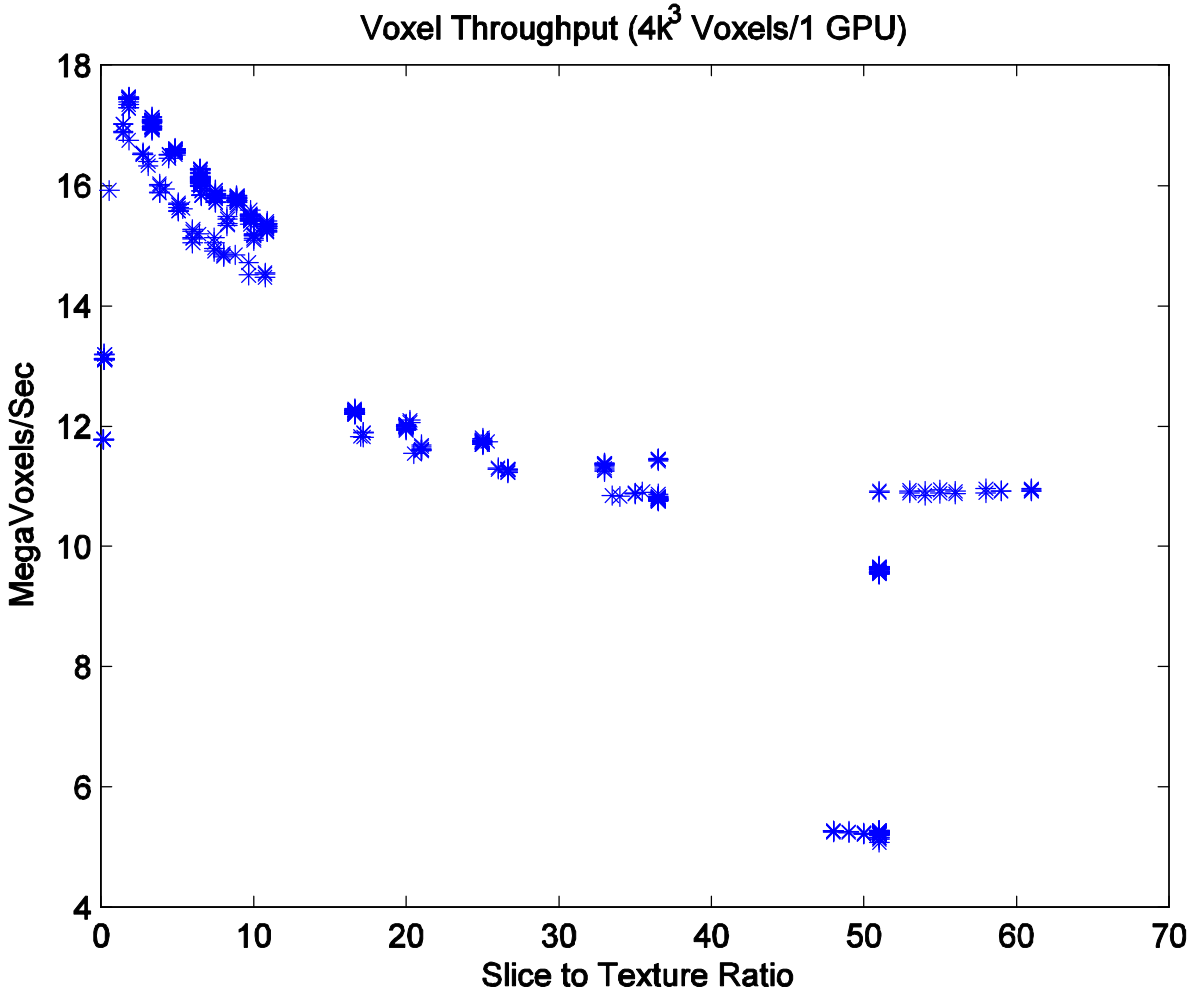
Evaluation

Sandia National Laboratories

- Supermicro workstation
  - Dual Octo-core Intel Xeon E-2687W @ 3.1 GHz w/ hyper threading
  - 512 GB RAM
  - 4 PCI-E 2.0 x16 slots
- 2 NVidia S2090 Devices
  - 4 Tesla M2090 GPUs each (8 total)
  - Connected via 4 PCI-E host interface cards
- M2090
  - 6 GB GDDR5 memory apiece
  - 16 streaming multiprocessors (SM)
  - 768 KB L2 Cache (load, store, and texture operations)
  - 32 Compute cores per SM
  - 48 KB L1 Memory (explicitly set, shared memory not used)
  - 8 KB Constant Memory and Texture Cache
- Two datasets tested
  - 64 Gigavoxels
  - 1 Teravoxel

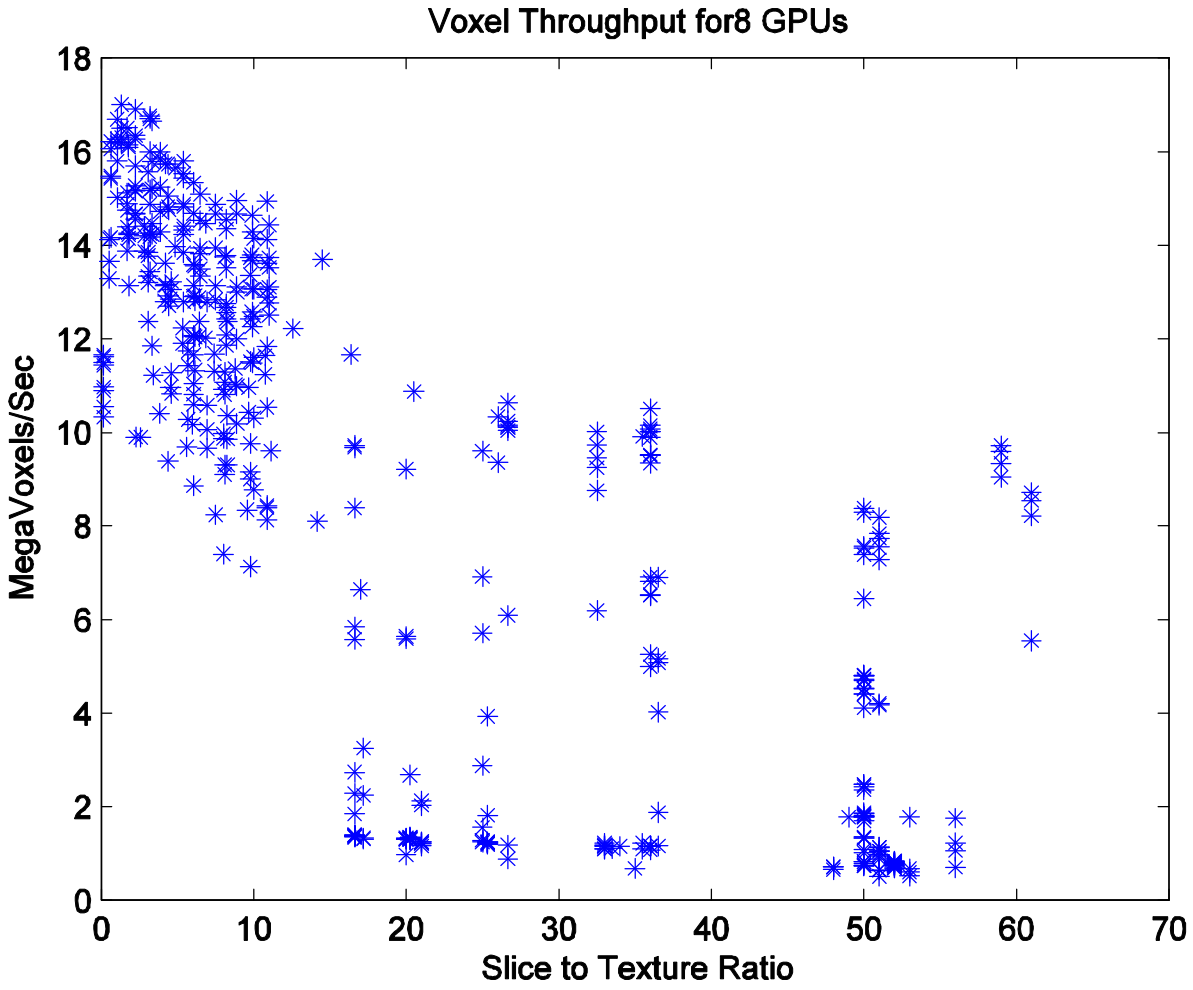
Results: Throughput 64 GV/ 1 GPU

Sandia National Laboratories



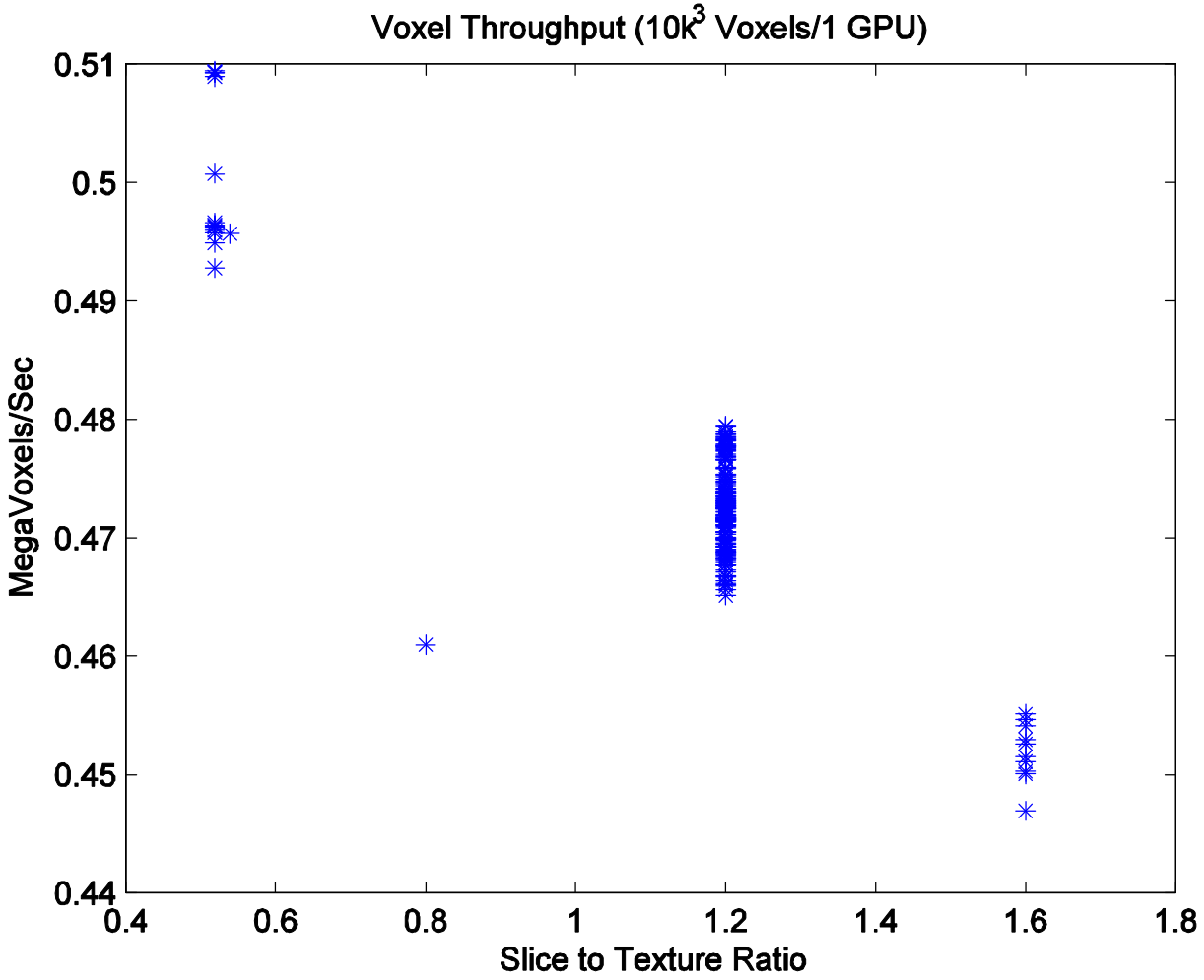
Results: Throughput 64 GV/ 8 GPUs

Sandia National Laboratories



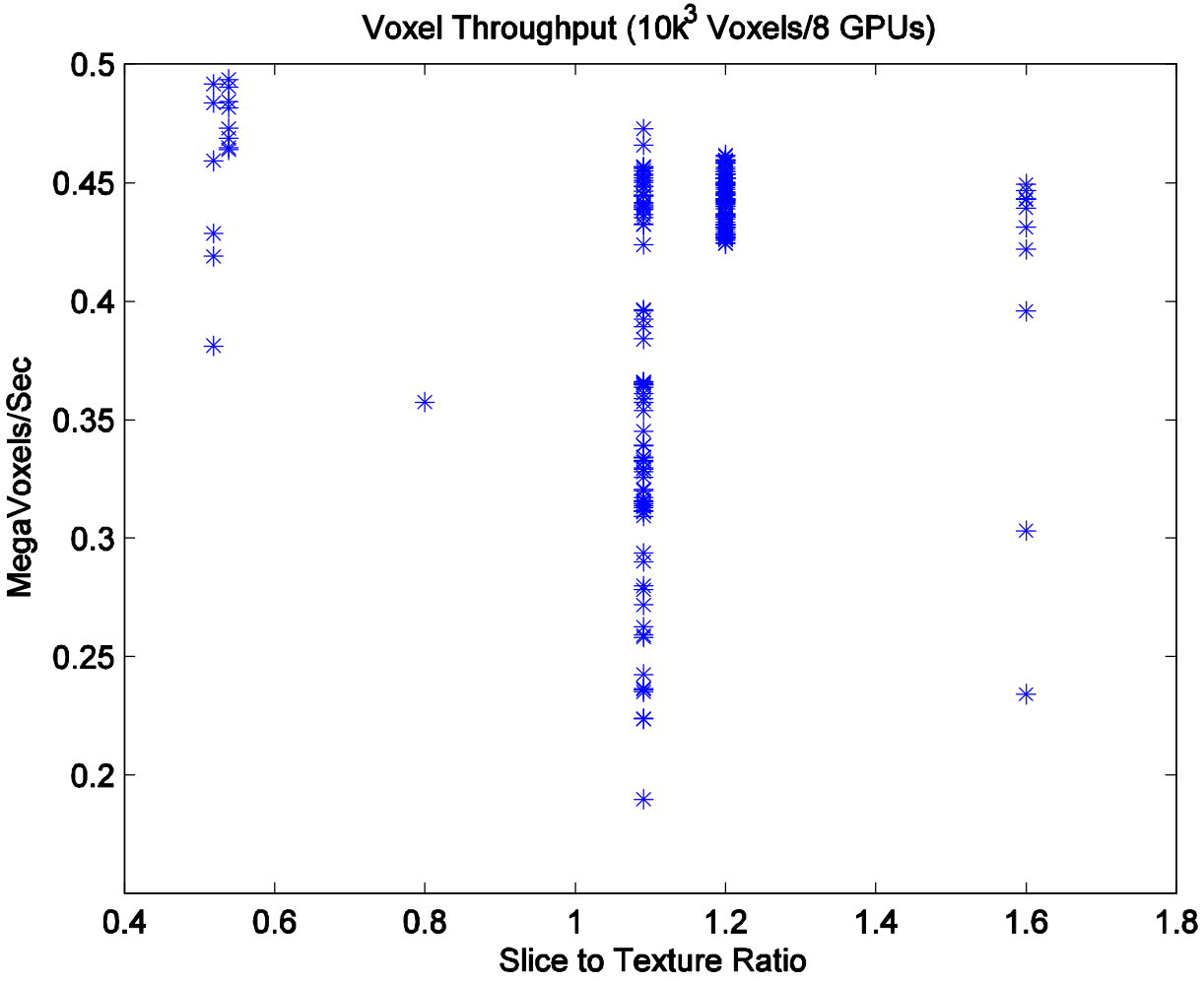
Results: Throughput 1 TV/ 1 GPU

Sandia National Laboratories



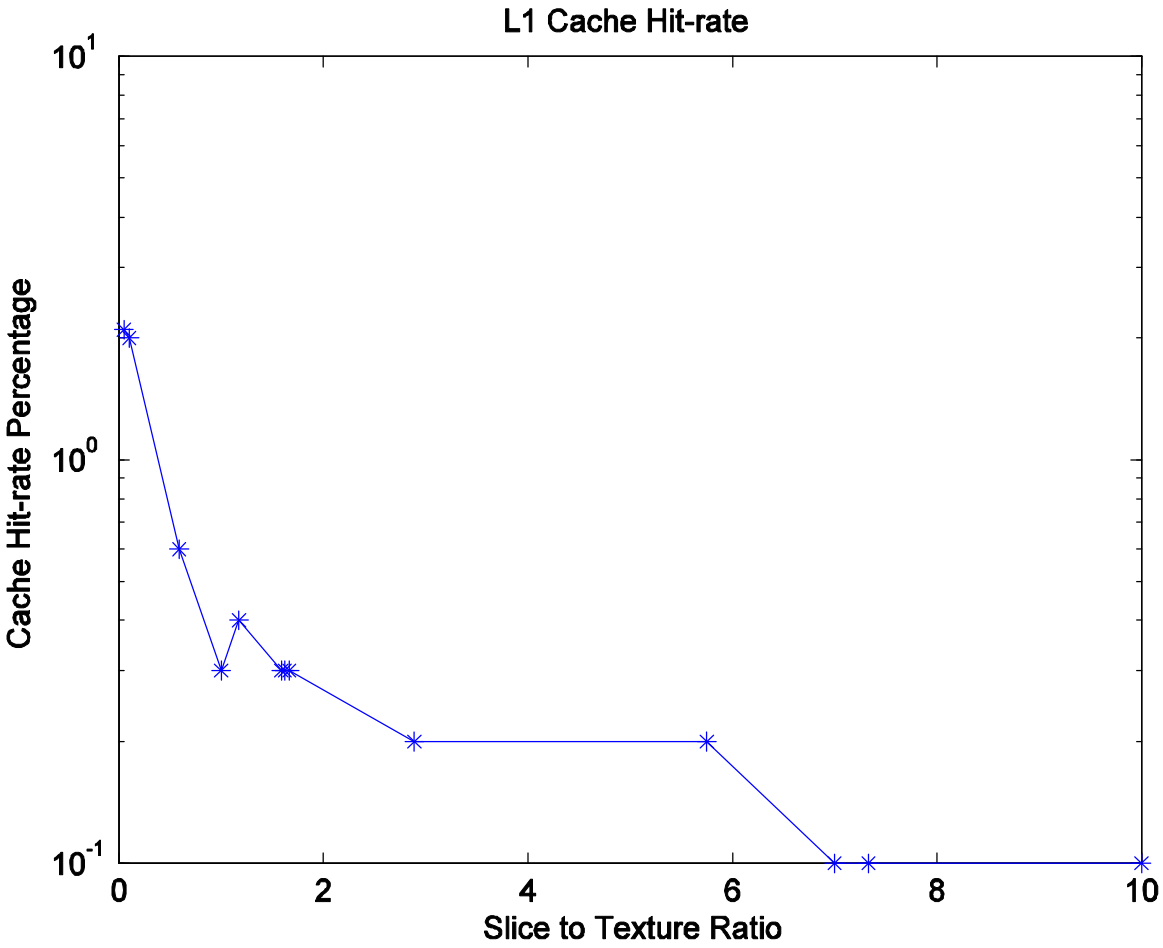
Results: Throughput 1 TV/ 8 GPUs

Sandia National Laboratories

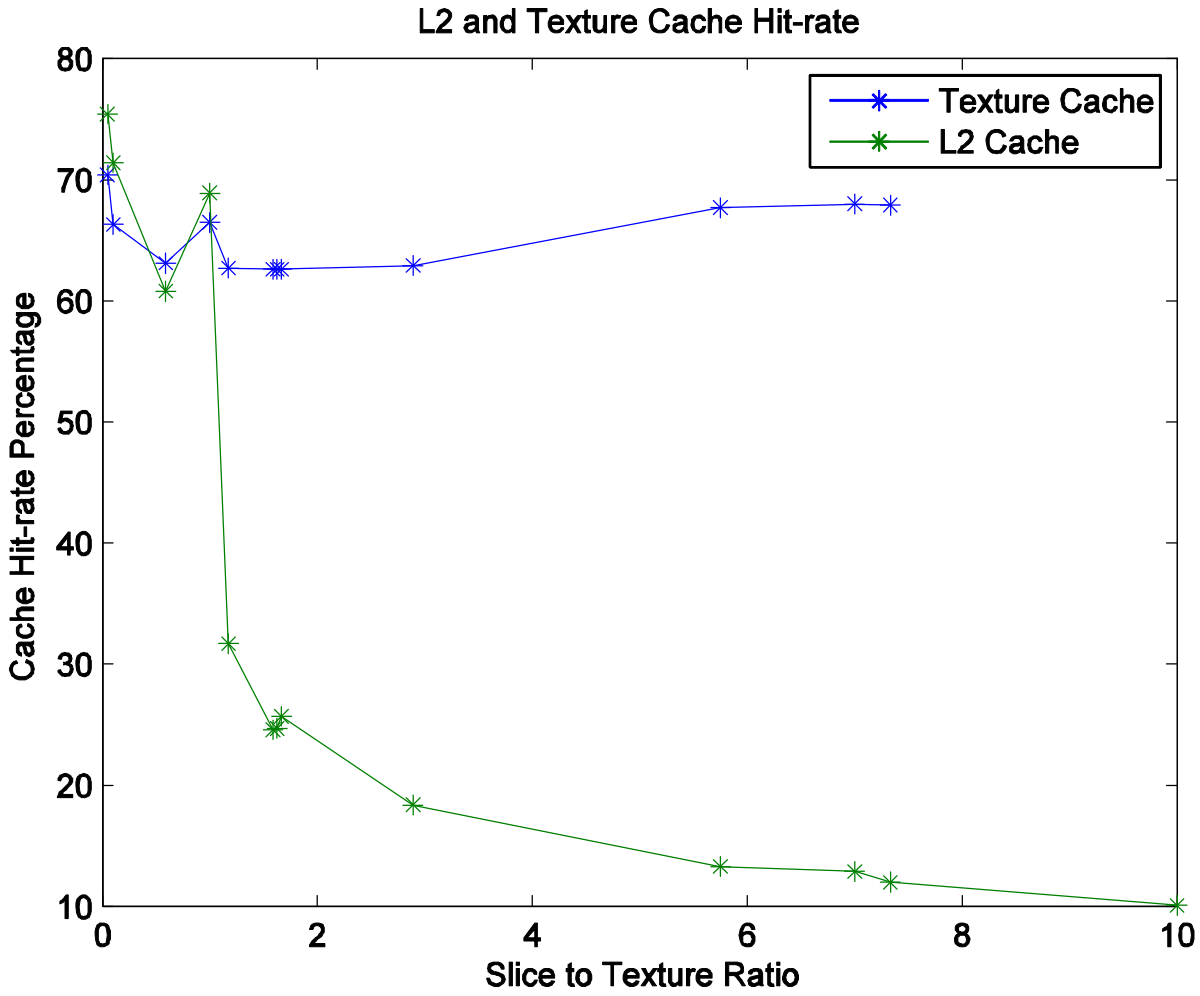


Results: L1 Cache Hit-rates

Sandia National Laboratories



Results: L2 and Texture Cache Hit-rates



Conclusion on Kernel Design

Sandia National Laboratories

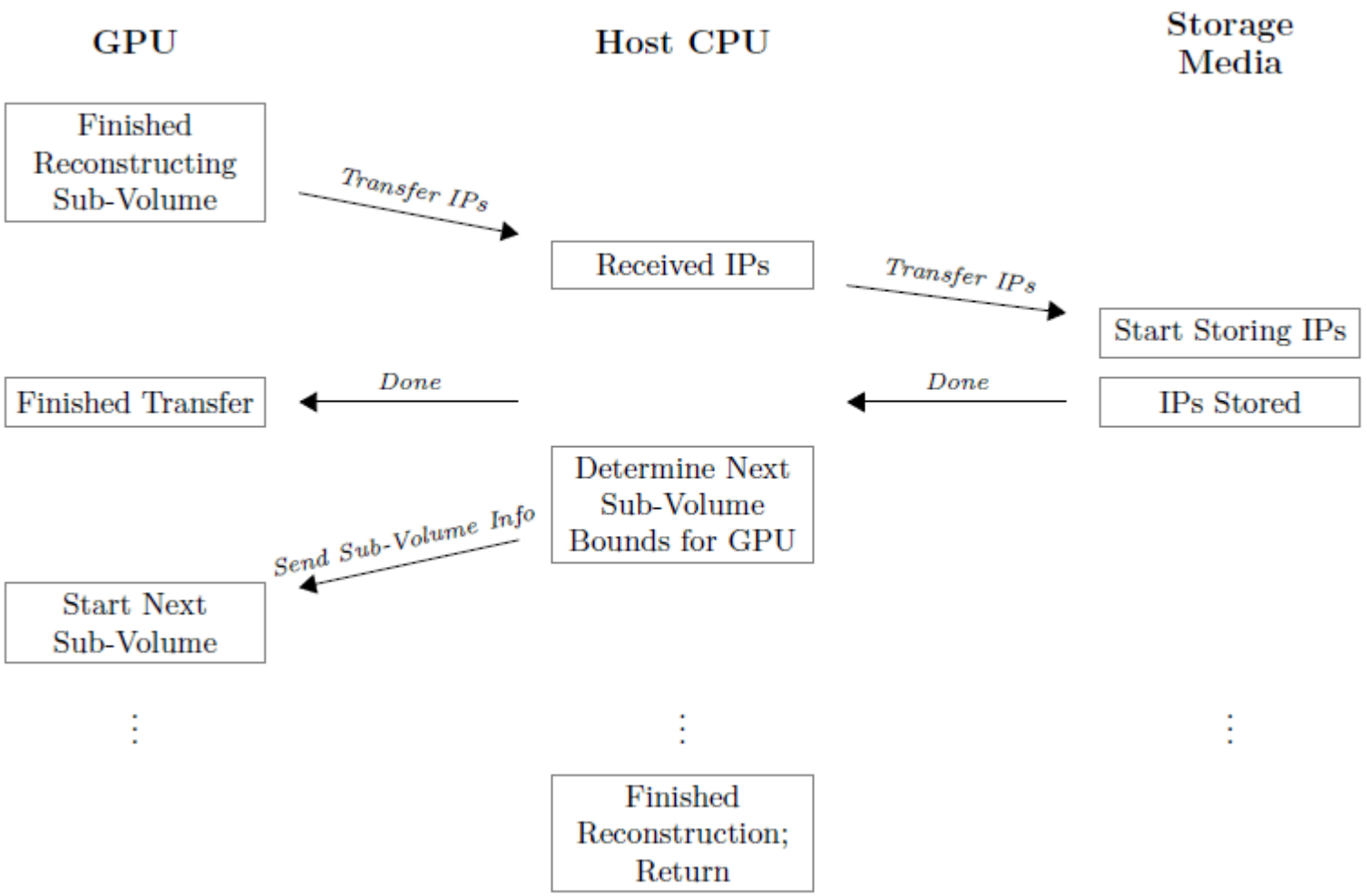
- Big Data CT Reconstruction kernels clearly benefit from an Irregular approach
  - Massive parallelism has potential to destroy spatial locality
  - Counter Intuitive approach may create performance gains
  - Irregular approach improves voxel throughput by improving cache-hit rates
  - Small X-ray data batches and large subvolume tend to perform best.
- Unfortunately this is only a portion of the problem
- This is a Big Data problem
  - Moving information to and from storage media
  - Host-side data
- Next Problem: Rest of the system cannot keep up.

Big Data CT

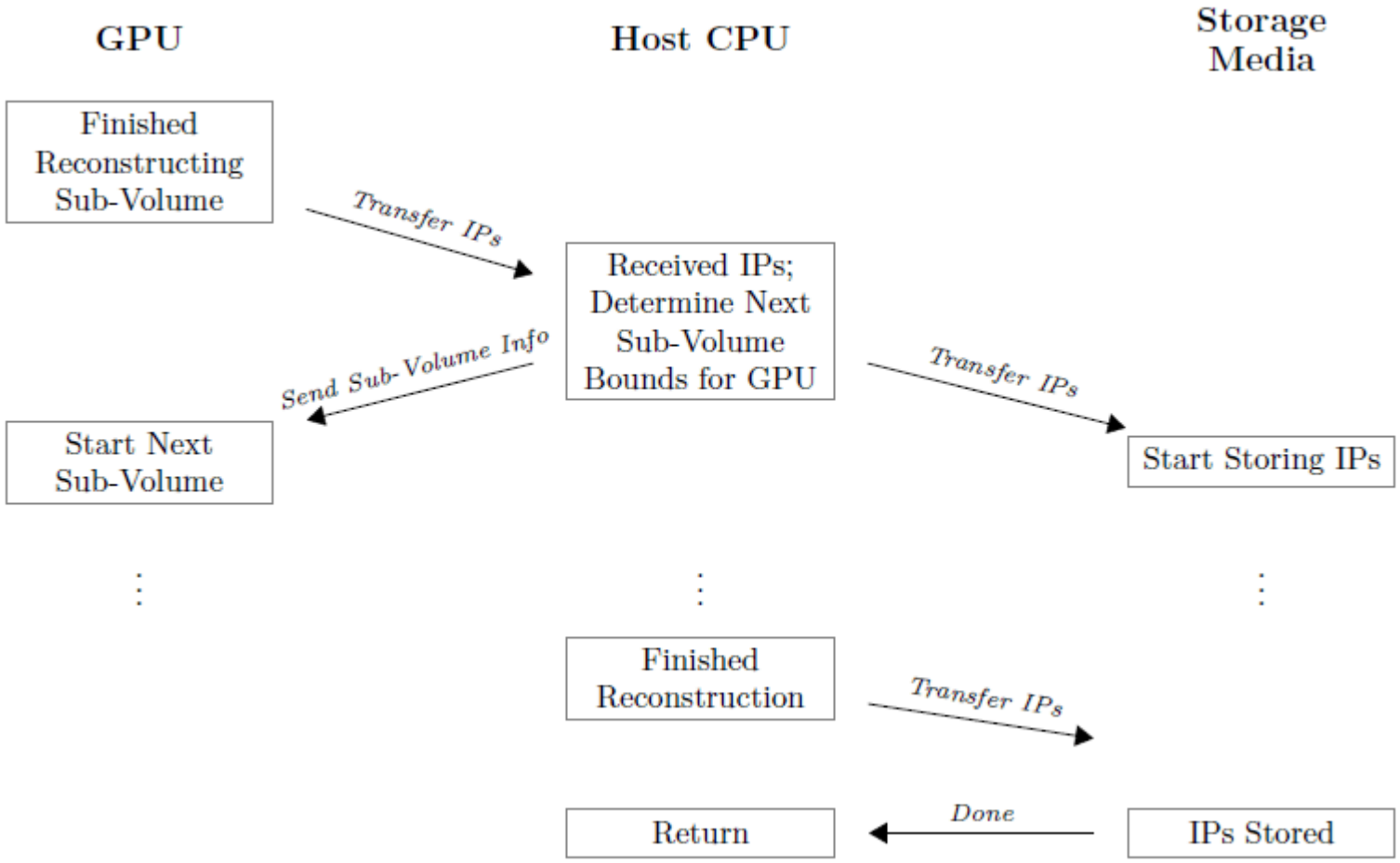
Sandia National Laboratories

- Industrial-Scale Computed Tomography Datasets
  - Differ from Medicine
  - Medical datasets are typically less than a few gigabytes
  - Industrial scale datasets are dozens of Gigabytes to several Terabytes!
- The reconstruction is large
  - The fastest supercomputers cannot be used due to memory.
  - Waiting days to years is not practical!
  - Suboptimal GPU utilization.
- Industrial-Scale CT is really Big Data CT that impacts many areas.
  - Non-Destructive Testing
  - Materials Characterizations
  - Quality Assurance
  - Verification and Validation

Serial Implementation



Modularization



Results: GPU Speedups

Sandia National Laboratories

- Speedup is relative to single thread CPU time of 44,987 hours

Number GPUs	Modular Time	Modular Speedup	Serial Time	Serial Speedup
1	102.7	438x	113.9	395x
2	53.3	844x	59.8	752x
4	30.3	1486x	37.4	1204x
6	23.1	1946x	28.0	1608x
8	19.9	2260x	23.3	1927x

Results: GPU Speedups

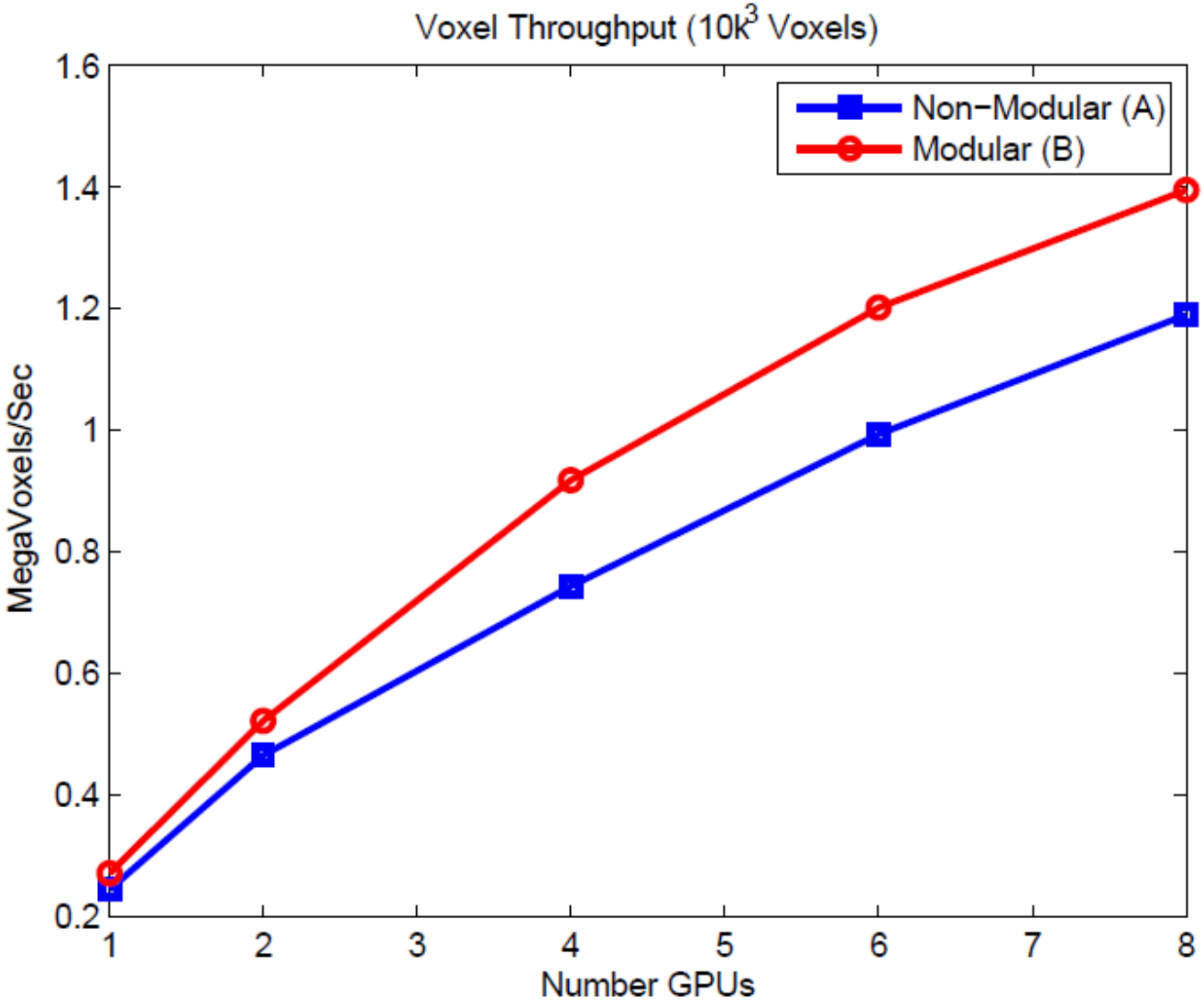
Sandia National Laboratories

- Speedup is relative to OpenMP CPU time of 4,039 hours

Number GPUs	Modular Time	Modular Speedup	Serial Time	Serial Speedup
1	102.7	39x	113.9	35x
2	53.3	76x	59.8	68x
4	30.3	133x	37.4	108x
6	23.1	175x	28.0	144x
8	19.9	203x	23.3	173x

Overall Improvements

Sandia National Laboratories



Pulling It All Together

Sandia National Laboratories

- GPUs are fast.
  - New Bottlenecks are cropping up.
  - Modularization helps.
- There has been little work in energy efficiency.
  - Are we trading one problem for another?
  - Most facilities have limited energy capacity
- For Big Data, consumption metrics may be unsustainable.
  - Same hindrance as Exascale Computing
- CPU-based vs. GPU-based comparisons
  - Computation times are proven, what about efficiency?
- Serial-based vs. Modularized-based comparisons
  - Does our approach help or hinder consumption?

Energy Efficiency

Sandia National Laboratories

- What aspects contribute the most?
  - Data transfer
  - Computation
  - I/O
- Looking at overall energy consumption is insufficient.
  - Does not factor performance.
- Other popular metrics:
  - Performance-per-Watt: Factors in power of the system.
  - Energy-Delay: Greater emphasis on performance.
    - Defined as *KWH*H.*
    - Lower is better.
- Studying a single metric can be misleading!

Overall Energy Consumption

Sandia National Laboratories

- Total energy used by an algorithm.
  - Simple metric to grasp.
- Comparison between systems and algorithms is straightforward.
- However, this metric does not factor the time consumed during the computation.
- Identical consumption is possible with dramatically different computation times

Performance-Per-Watt

Sandia National Laboratories

- This metric is measured as “million-instructions-per-second-per-watt”.
- Somewhat factors computation time
- This metric is typically used when energy consumption is vital.
  - Systems that run on primarily batteries (e.g. notebooks, tablets, etc.)
- Not sufficient as one could sacrifice speed to increase this metric.

Energy-Delay

Sandia National Laboratories

- The Energy-Delay metric is a relatively new metric.
  - Total Energy (KWH) multiplied by computation time.
- Some forms of the energy delay product square or cube the delay when greater emphasis on performance is desired.
- First proposed to evaluate trade-offs between circuit level power savings techniques for digital designs.

Implementation

Sandia National Laboratories

- Leveraging pinned host memory minimizes idling between kernel launches applied to a given sub-volume.
- As mentioned, data writing is another source of GPU idle.
  - Ideally, the modularized approach would minimize this idling.
  - Is this savings moot with the complexity of the task on the CPU?
- Both implementations are written in a hybrid environment:
  - C++
  - CUDA (Ver. 5.0)
  - OpenMP

Evaluation

Sandia National Laboratories

- Metrics to be measured:
  - Energy Consumption
    - To be measured in kilowatt-hours (kWh).
  - Performance-Per-Watt
    - Presented as voxels-reconstructed (and stored)-per-second-per-watt.
    - Modified from traditional MFLOPS-per-watt
    - Reconstruction is known to be limited in memory bandwidth and computation.
  - Energy-Delay Product
    - To ensure algorithm is not trading off speed for savings.
    - Square weighting of delay as suggested by other works (Laros III et. Al.).

Evaluation Cont.

Sandia National Laboratories

- Four CT implementations measured:
  - Popular CPU-based Approach
    - Leverages OpenMP for multi-threading and MPI for parallel processing
  - Naïve GPU Implementation:
    - Baseline
    - No GPU-specific exploitation
    - Brute force Port of CPU-like implementation
    - Up to 8 GPUs
  - Serialized Approach
    - Up to 8 GPUs
  - Modularized Approach
    - Up to 8 GPUs

Evaluation Cont.

Sandia National Laboratories

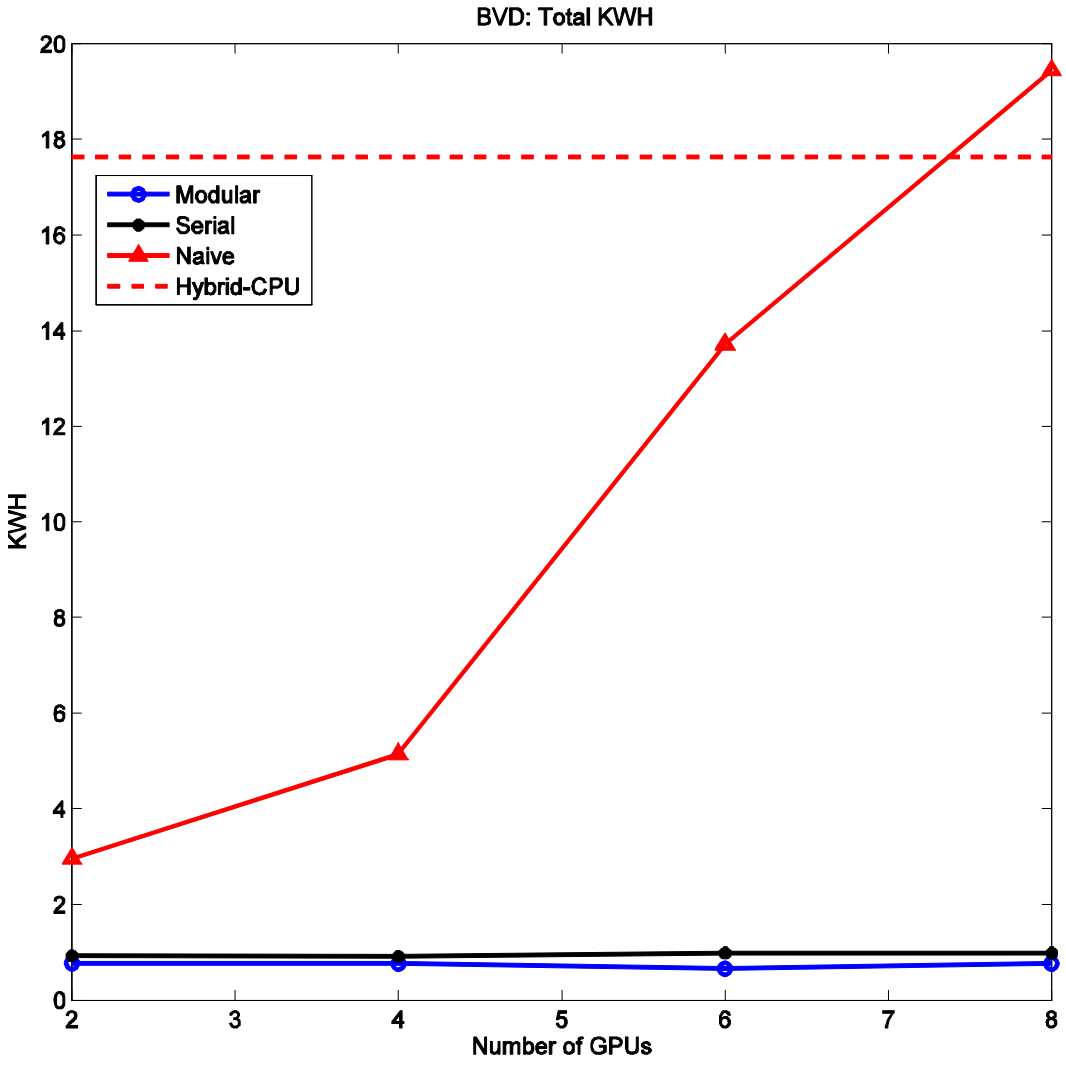
- System used:
  - Supermicro server
    - Dual octo-core Intel Xeon E-2687W @ 3.1GHz w/ Hyper threading
    - 512GB system memory
  - 8 NVidia Tesla M2090 GPUs
    - Contained in 2 NVidia S2090 units
    - Connected via 4 PCI-E 2.0 x16 host interface cards
  - Storage Media
    - 8 x 3TB 6 Gb/s drives in RAID 0
    - Controlled by an Intel Controller with 1 GB of DDR3 cache.
  - Metrics obtained with several P3 International P4460 Kill-A-Watt Monitors
  - GPUs disconnected for CPU measurements

Evaluation Cont.

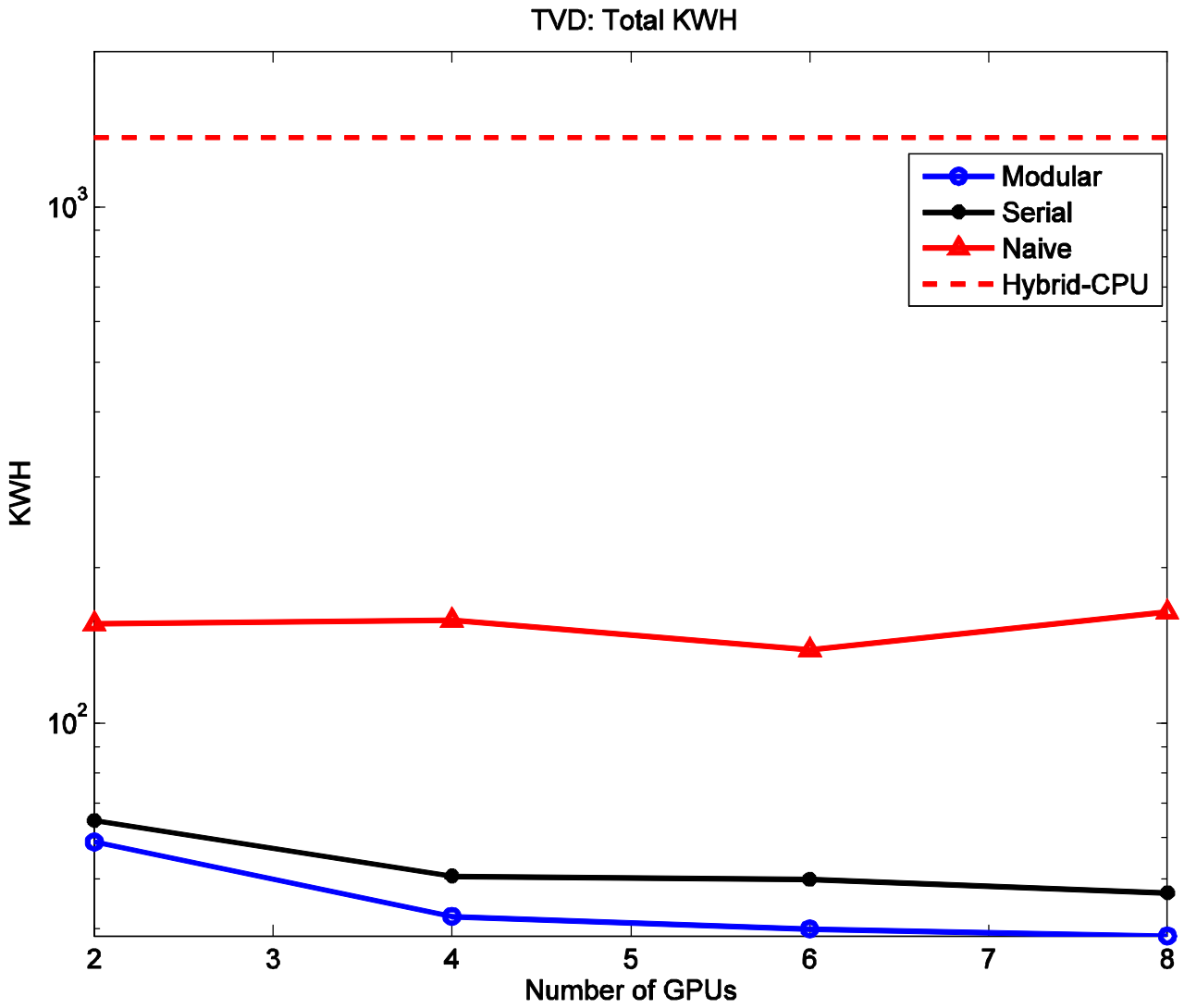
Sandia National Laboratories

- Two datasets used
- 64 Gigavoxel Reconstruction
  - 4000x4000x4000 voxel volume
  - 1800 Projections
  - High-end of current acquisitions
- 1 Teravoxel Reconstruction
  - 10K x 10K x 10K voxel volume
  - 10,000 Projections
  - “Future-proof” dataset.

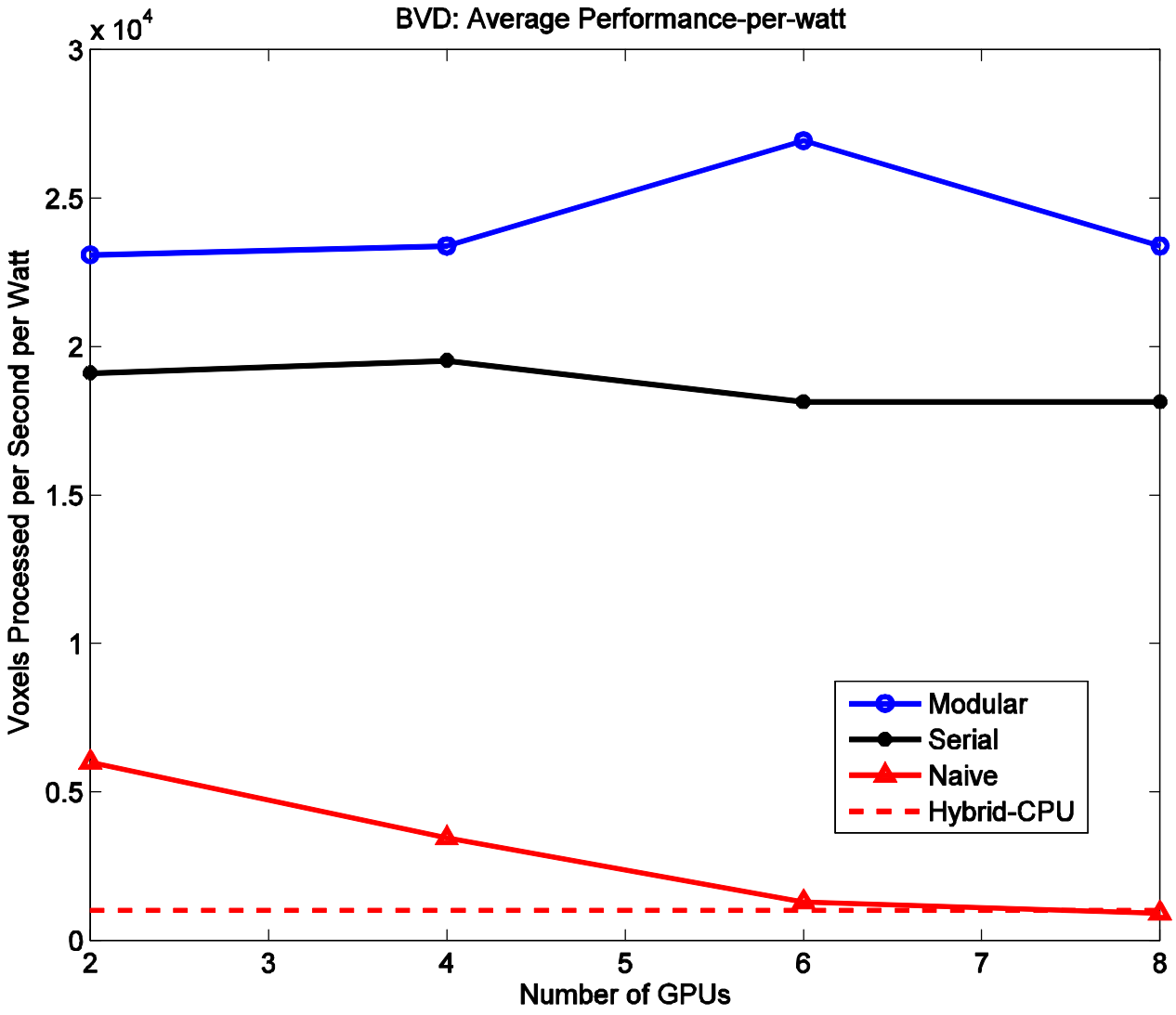
Energy Consumption 64 GV



Energy Consumption 1 TV

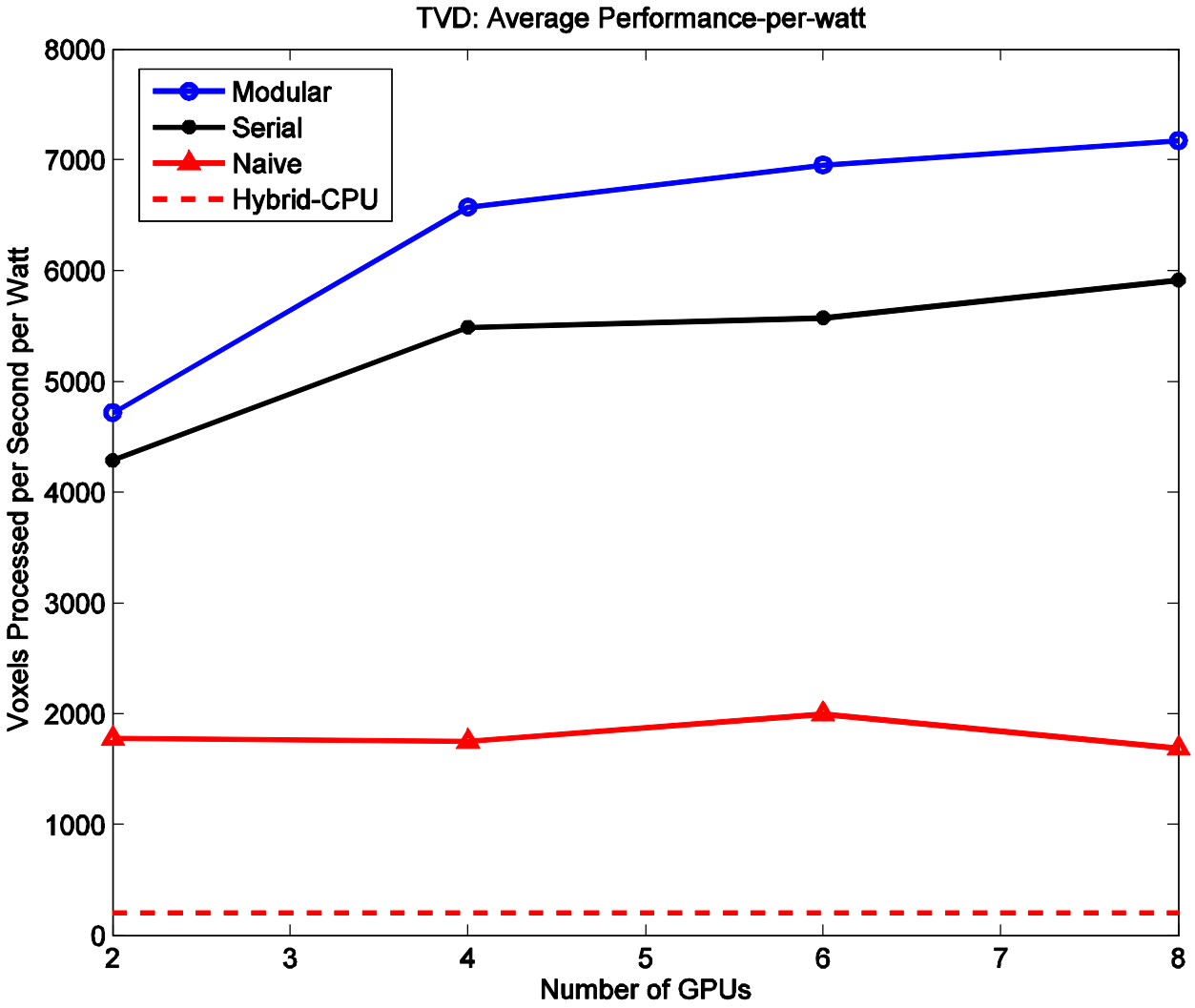


Performance-Per-Watt 64 GV

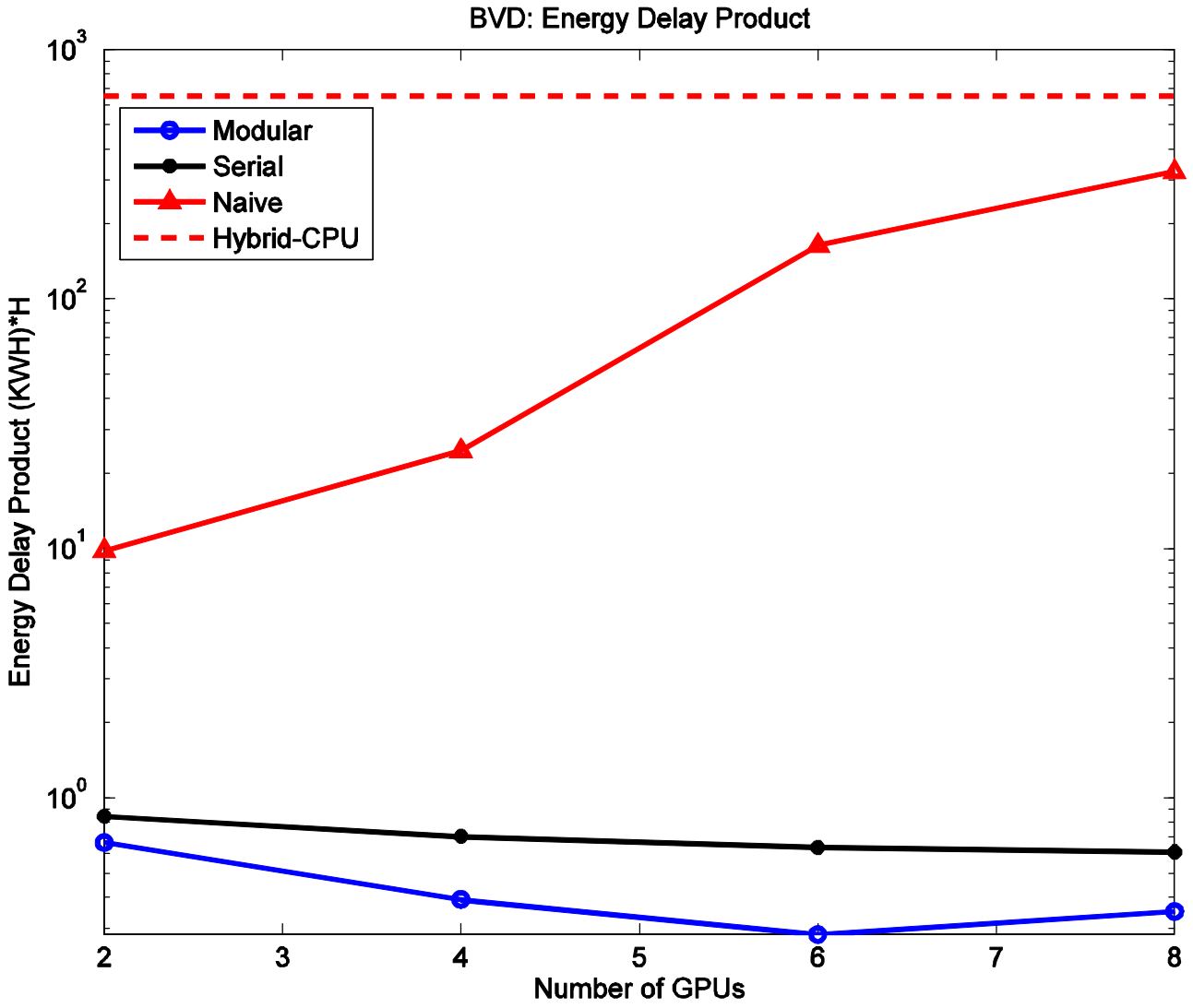


Performance-Per-Watt 1 TV

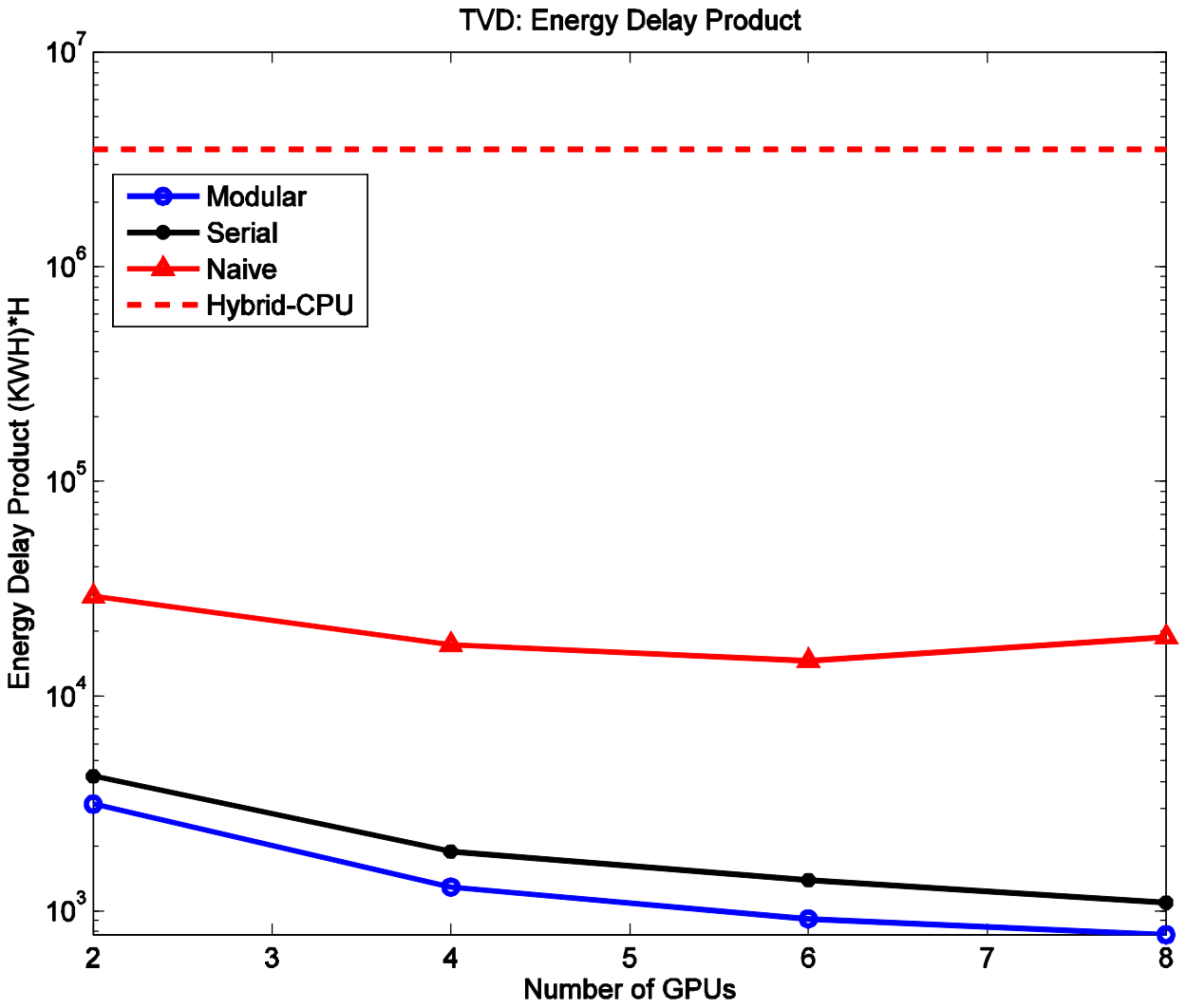
Sandia National Laboratories



Results: Energy-Delay 64 GV



Results: Energy-Delay 1 TV



Conclusions

Sandia National Laboratories

- This work has shown that leveraging GPUs not only improves computation time, but does so while improving multiple energy metrics for FDK reconstruction.
- The computation community is approaching energy limitations, intelligent algorithm design can help.
- Testing a heterogeneous cluster approach with promising results!
- What other algorithms can benefit from this work?